Data Security

Lecture No. (9)

Dr/ Roayat Ismail

Outlines:

We have three main protection mechanisms:

1. Cryptography
2. Access control
3. Firewalls

1- Access Control

- It is the prevention of unauthorized use of a resource.
- Entity authentication deals with the problem of determining whether a user should be allowed access to a particular system or resource.
- The entity whose identity needs to be proved is called the claimant; the party that tries to prove the identity of the claimant is called the verifier. When Bob tries to prove the identity of Alice, Alice is the claimant, and Bob is the verifier.

1- Access Control

It involves two processes:

1. Identification
2. Authentication

1. Identification

- It is a mechanism that provides information about an unverified entity-called the claimant -that wants to be granted access to a known entity. The label applied to the claimant is called an identifier (ID). The identifier must be a unique value (random numbers, or special characters.)

2. Authentication

- It is the process of validating a claimant's identity. It ensures that the entity requesting access is the entity claimed.

What is the difference between:

1. Message authentication
2. Entity authentication

- There are two differences between message authentication and entity authentication. First, message authentication may not happen in real time; entity authentication does. In the former, Alice sends a message to Bob. When Bob authenticates the message, Alice may or may not be present in the communication process. On the other hand, when Alice requests entity authentication, there is no real message communication involved until Alice is authenticated by Bob. Alice needs to be online and takes part in the process. Only after she is authenticated can messages be communicated between Alice and Bob.

- Second, message authentication simply authenticates one message; the process needs to be repeated for each new message. Entity authentication authenticates the claimant for the entire duration of a session.

There are four types of (entity) authentication mechanisms:

1. Something you know
2. Something you have (token)
3. Something you are (static biometrics)
4. Something you produce
(dynamic biometrics)

1. Something you know:

password, passphrase or other unique authentication code such as PIN (Short for **Personal Identification Number**, **PIN** is a set of personal numbers used to prove positive identification. Often used with automated bank teller machines, accessing wireless networks.

2. Something you have:

a card (includes ATM cards with magnetic strips containing PIN against which user input is compared), a passport, a driver's license, an identification card, a credit card.

3. **Something you are**:

Something inherent in the user that is evaluated using biometric as:

Fingerprints- Facial recognition- hand geometry- retina recognition – Iris recognition.

4- **Something you produce:**

Signature- voice pattern:(it captures the analog waveform of human speech and compare it to a stored version- it provide the user with a phrase that they must read.

We will consider in more details some examples on “Something you know”

entity authentication

1.Password based Authentication

- Transmit password in clear over the network to the server
- Main Problem
  - Eavesdropping/Interception

Passwords Authentication

- The simplest and the oldest method of entity authentication is the password, something that the claimant possesses. A password is used when a user needs to access a system to use the system's resources (log-in). Each user has a user identification that is public and a password that is private. We can divide this authentication scheme into two separate groups: the fixed password and the one-time password.

Fixed Password:

- In this group, the password is fixed; the same password is used over and over for every access. This approach is subject to several attacks.
- Eavesdropping: Eve can watch Alice when she types her password. Most systems, as a security measure, do not show the characters a user types.
- Eavesdropping can take a more sophisticated form. Eve can listen to the line and then intercept the message, thereby capturing the password for her own use.

- Stealing a Password. The second type of attack occurs when Eve tries to physically steal Alice's password. This can be prevented if Alice does not write down the password; instead, she just commits it to memory. Therefore, a password should not be very simple nor related to something familiar to Alice.
- Accessing a file. Eve can hack into the system and get access to the file where the passwords are stored.
- Guessing. Eve can log into the system and try to guess Alice's password by trying different combinations of characters.

- A more secure approach for storing a password:

It is to store the hash of the password in the password file (instead of the plaintext password). Any user can read the contents of the file, but, because the hash function is a one-way function, it is almost impossible to guess the value of the password. The hash function prevents Eve from gaining access to the system even though she has the password file. However, there is a possibility of another type of attack called the dictionary attack.

In this attack, Eve is interested in finding one password, regardless of the user ID. For example, if the password is 6 digits, Eve can create a list of 6-digit numbers (000000 to 999999), and then apply the hash function to every number; the result is a list of 1 million hashes. She can then get the password file and search the second-column entries to find a match. This could be programmed and run offline on Eve's private computer. After a match is found, Eve can go online and use the password to access the system. We can overcome this type of attack by increasing the length of the password.

- Another approach to increase the security:

two identification techniques are combined. A good example of this type of authentication is the use of an ATM card with a PIN (personal identification number). The card belongs to the category "something you have" and the PIN belongs to the category "something you know." The PIN is actually a password that enhances the security of the card. If the card is stolen, it cannot be used unless the PIN is known. The PIN, however, is traditionally very short so it is easily remembered by the owner. This makes it vulnerable to the guessing type of attack.

One-Time Password

- In this type of scheme, a password is used only once. It is called the one-time password. A one-time password makes eavesdropping and stealing useless. However, this approach is very complex.

2. Cryptographic Authentication

- No password or secret is transferred over the network
- Users prove their identity to a service by performing a cryptographic operation, usually on a quantity supplied by the server.
- Crypto operation based on user's secret key

2.1 **Challenge-Response Authentication**

(Something you know)

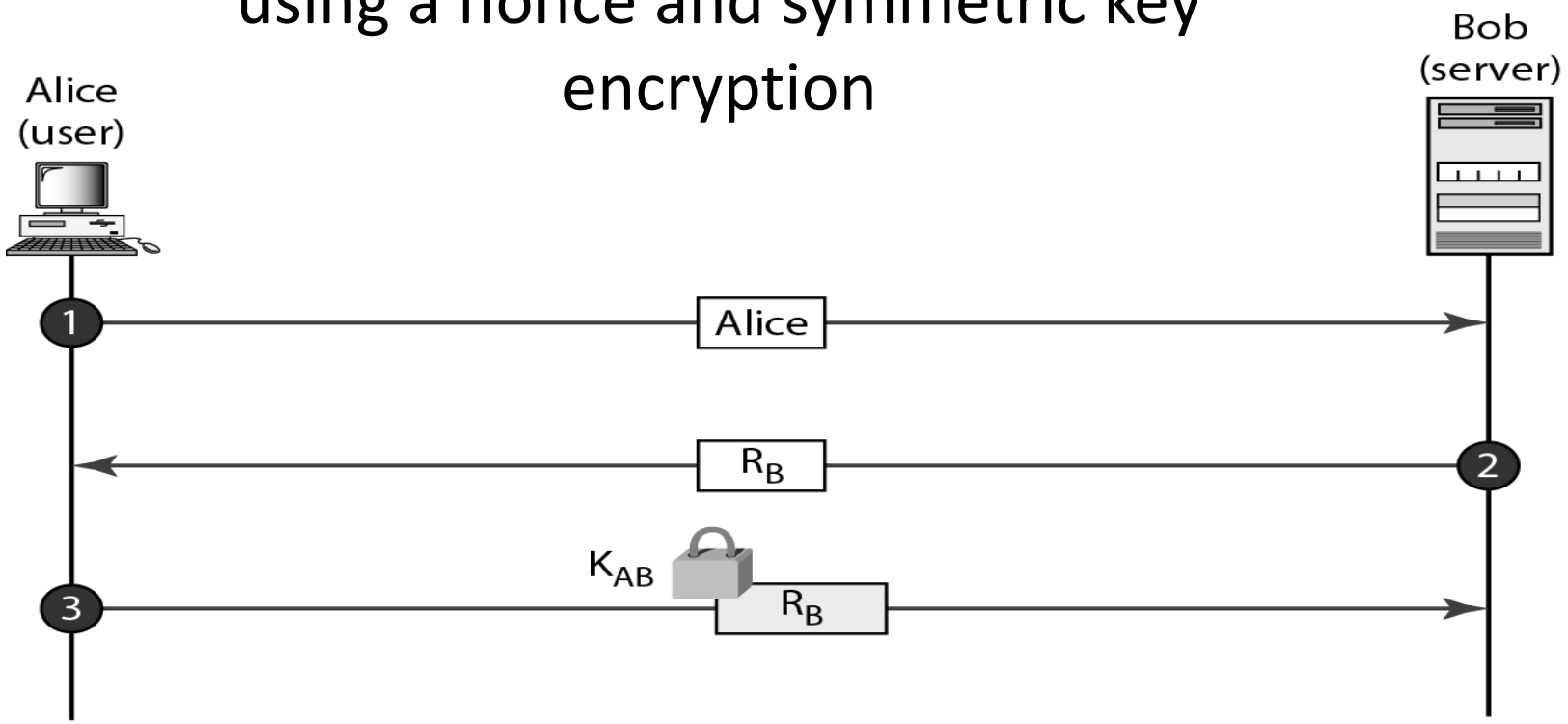
- In password authentication, the claimant proves her identity by demonstrating that she knows a secret, the password. However, since the claimant reveals this secret, the secret is susceptible to interception by the adversary. In challenge-response authentication, the claimant proves that she knows a secret without revealing it. In other words, the claimant does not reveal the secret to the verifier; the verifier either has it or finds it.

- The challenge is a time-varying value such as a random number (nonce) or a timestamp which is sent by the verifier. The claimant applies a function to the challenge and sends the result, called a response, to the verifier. The response shows that the claimant knows the secret.

1- Using a Symmetric-Key Cipher:

- In the first category, the challenge-response authentication is achieved using symmetric key encryption. The secret here is the shared secret key, known by both the claimant and the verifier. The function is the encrypting algorithm applied on the challenge

2.2-Challenge/response authentication using a nonce and symmetric key encryption

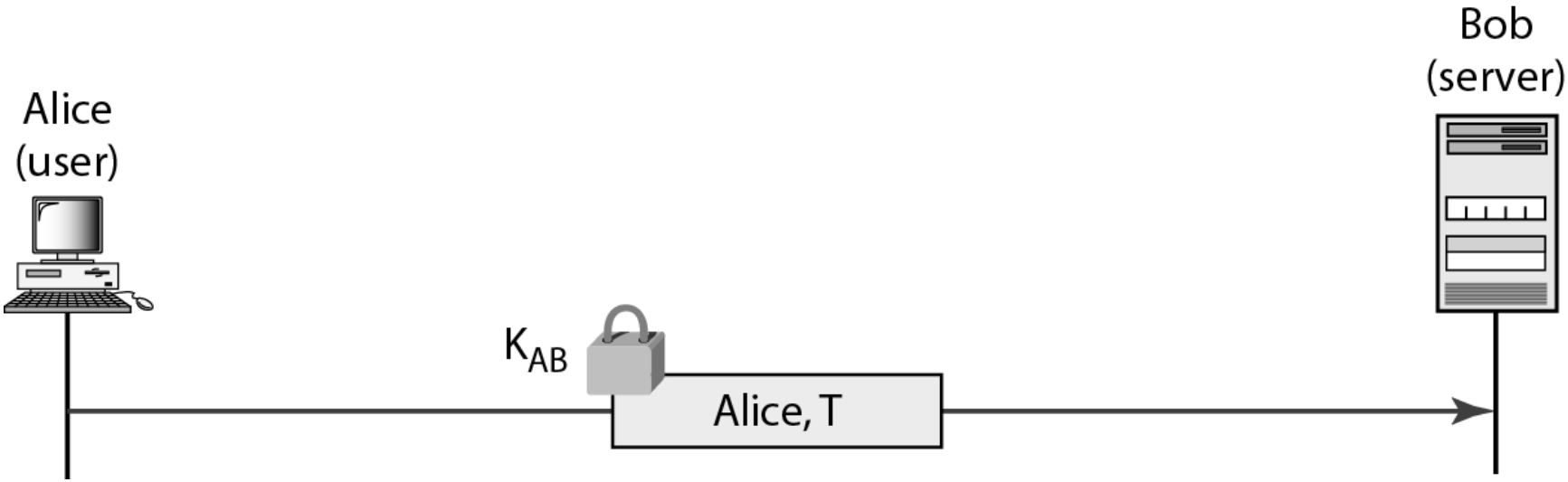


use of a nonce prevents a replay of the third message by Eve. Eve cannot replay the third message and pretend that it is a new request for authentication by Alice because once Bob receives the response; the value of RB is not valid any more. The next time a new value is used.

- The first message is not part of challenge-response; it only informs the verifier that the claimant wants to be challenged. The second message is the challenge. And RB is the nonce randomly chosen by the verifier to challenge the claimant. The claimant encrypts the nonce using the shared secret key known only to the claimant and the verifier and sends the result to the verifier. The verifier decrypts the message. If the nonce obtained from decryption is the same as the one sent by the verifier, Alice is granted access.

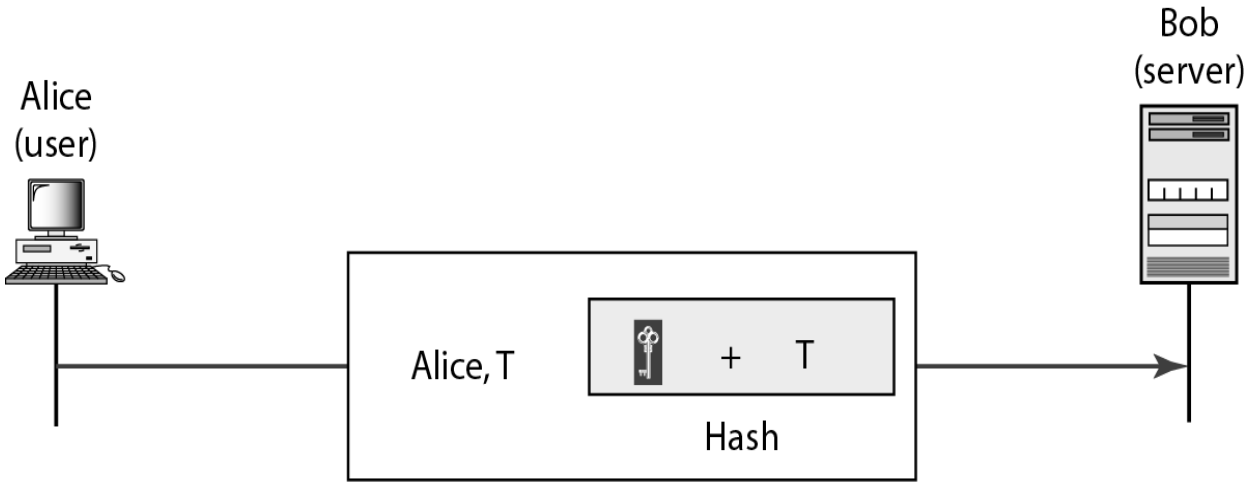
- In the second approach, the time-varying value is a timestamp, which obviously changes with time. In this approach the challenge message is the current time sent from the verifier to the claimant. However, this supposes that the client and the server clocks are synchronized; the claimant knows the current time. This means that there is no need for the challenge message. The first and third messages can be combined. The result is that authentication can be done using one message, the response to an implicit challenge, the current time. The following Figure shows the approach.

2.3-Challenge-response authentication using a timestamp and symmetric key encryption



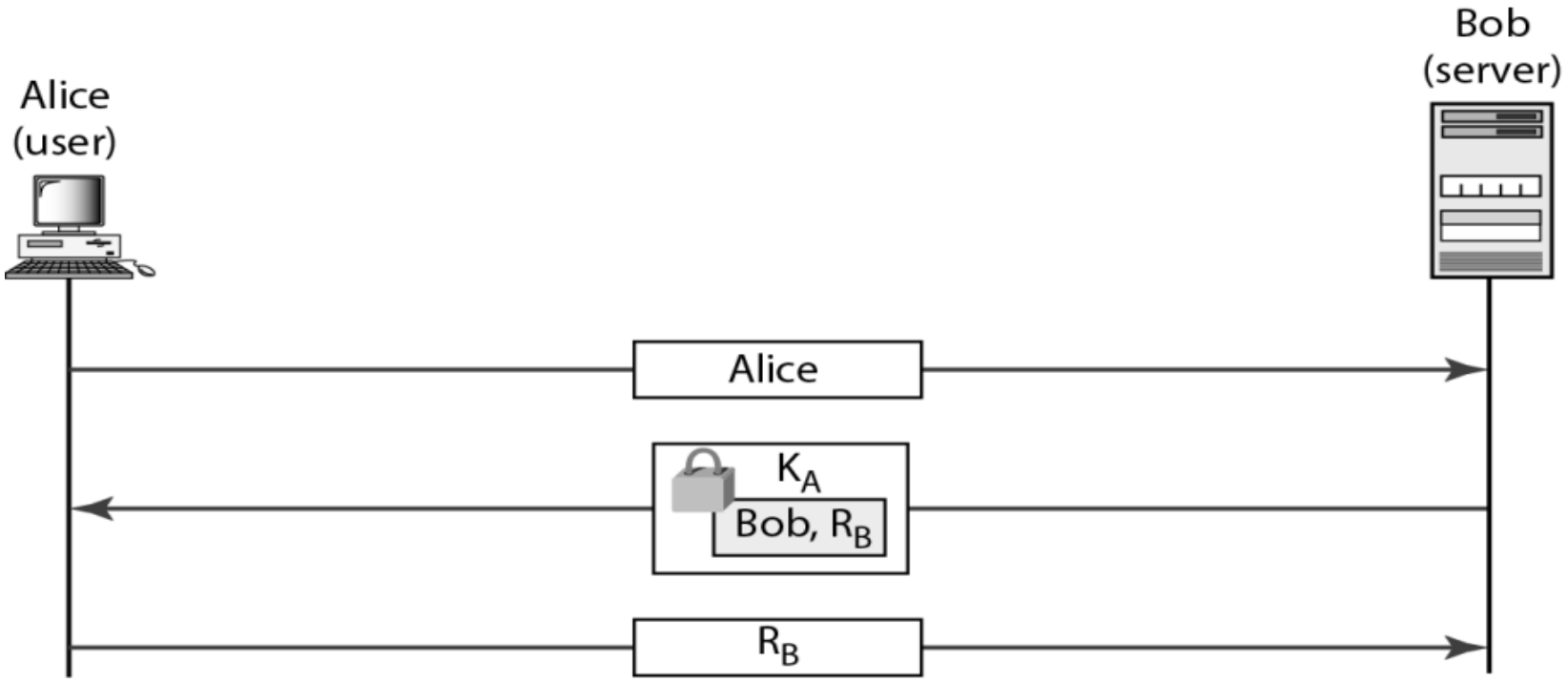
2.4- Challenge/response authentication Using Keyed-Hash Functions and timestamp (or a nonce)

- Instead of using encryption and decryption for entity authentication, we can use a keyed-hash function (MAC). There are two advantages to this scheme. In using a keyed-hash function, we can preserve the integrity of challenge and response messages and at the same time use a secret, the key.



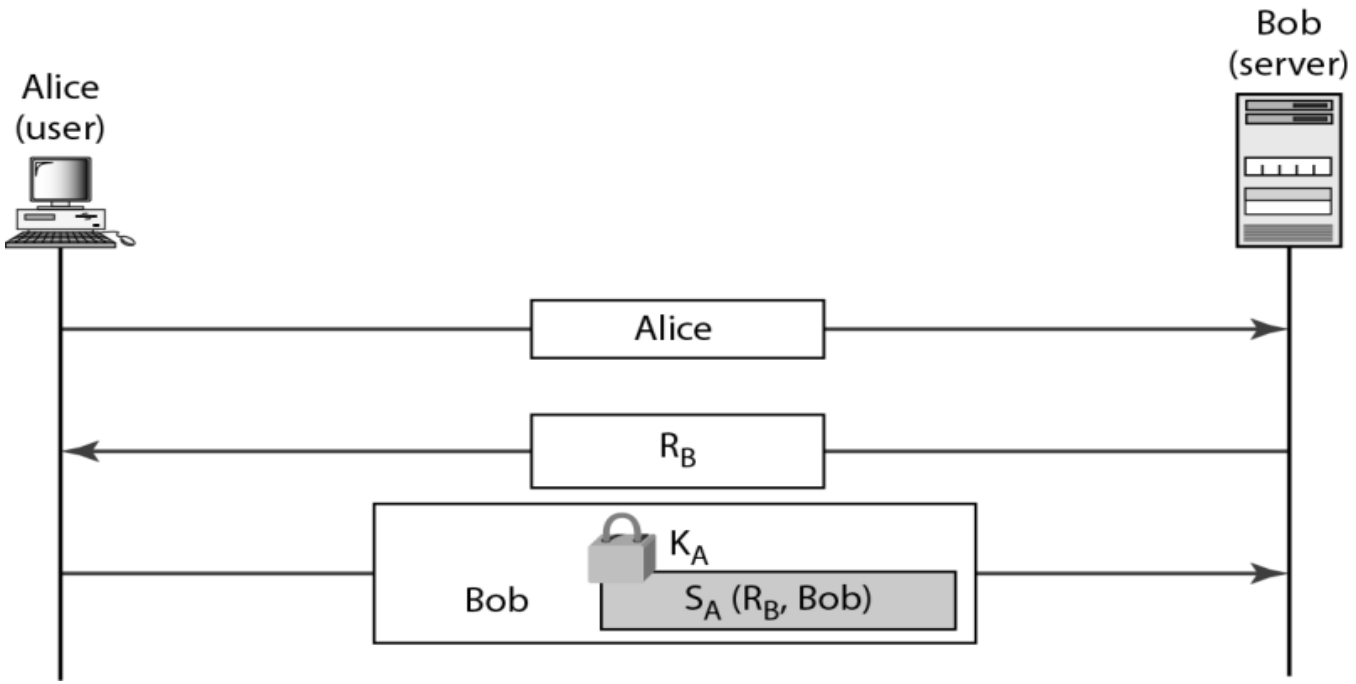
2.5- Challenge/response authentication using Asymmetric-Key Cipher and a nonce (or timestamp)

Instead of a symmetric-key cipher, we can use an asymmetric-key cipher for entity authentication



2.6 Challenge-response authentication Using Digital Signature and nonce (or timestamp)

We can use digital signature for entity authentication. In this method, we let the claimant use her private key for signing instead of using it for decryption.



3. Kerberos

What is Kerberos?

- Developed at M.I.T.
- A secret key based service for providing authentication in open networks
- Authentication mediated by a trusted 3rd party on the network:
  - Key Distribution Center (KDC)

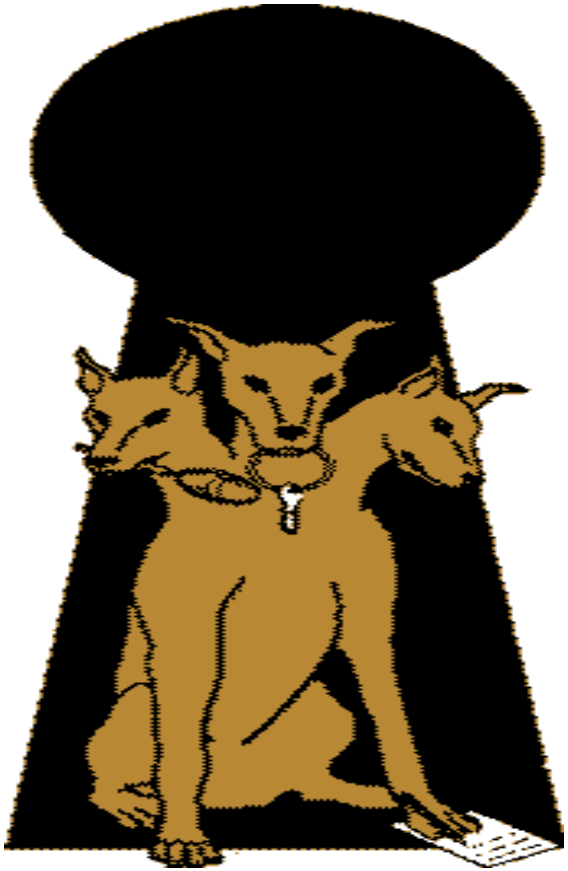
Kerberos: etymology

- The 3-headed dog that guards the entrance to Hades.
- Originally, the 3 heads represented the 3 A's

- Authentication

- Authorization

- Auditing (Login)



Some Kerberos benefits

- Standards based strong authentication system
- Wide support in various operating systems
- Make strong authentication readily available for use with campus computer systems
- Prevents transmission of passwords over the network
- Provides “single-sign-on” capability
  - Only 1 password to remember
  - Only need to enter it once per day (typically)

Kerberos is a network authentication protocol

- MIT took an idea from Xerox: “The Needham-Schroeder Protocol” which use Mediated Authentication

- Centralized, single sign-on, encrypted logins

Mediated Authentication

- A trusted third party mediates the authentication process
- Called the Key Distribution Center (KDC)
- Each user and service shares a secret key with the KDC
- KDC generates a session key, and securely distributes it to communicating parties
- Communicating parties prove to each other that they know the session key

Single Sign-On

1) I login to my desktop

2) After that initial login I'm given a ticket

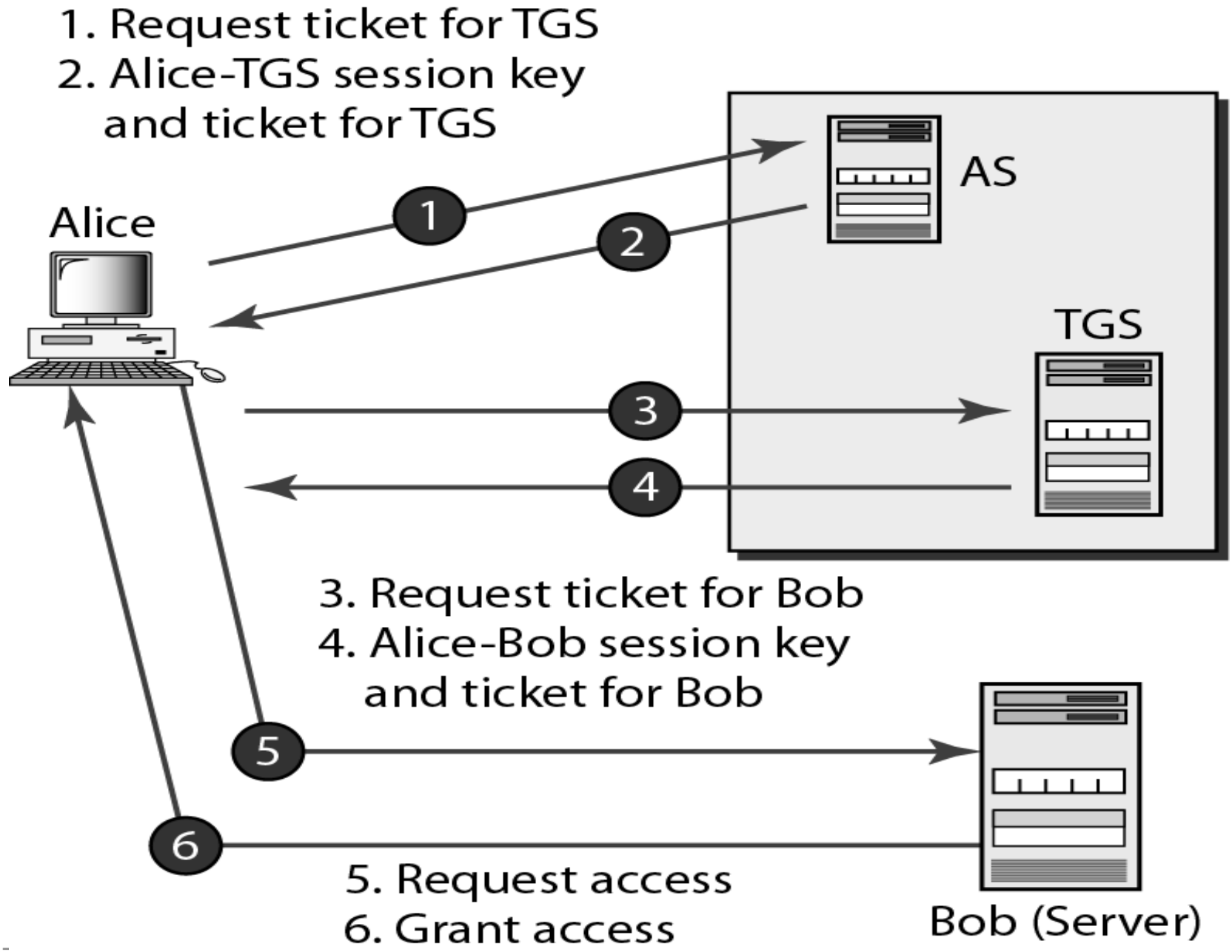
3) I can telnet to other machines on the network without typing a password again!

My password is not sent.

My ticket allows me to request more tickets to other services.

- Kerberos is an authentication protocol and at the same time a KDC that has become very popular. Several systems including Windows 2000 use it.

- Kerberos Servers: Three servers are involved in the Kerberos protocol: an authentication server (AS), a ticket-granting server (TGS), and a real (data) server that provides services to others. In our examples and figures Bob is the real server and Alice is the user requesting service. The following Figure shows the relationship between these three servers.


1. Request ticket for TGS
2. Alice-TGS session key
and ticket for TGS
AS
Alice
1
2
TGS
3
4
3. Request ticket for Bob
4. Alice-Bob session key
and ticket for Bob
5
6
5. Request access
6. Grant access
Bob (Server)

- **Authentication Server (AS)** AS is the KDC in Kerberos protocol. Each user registers with AS and is granted a user identity and a password. AS has a database with these identities and the corresponding passwords. AS verifies the user, issues a session key to be used between Alice and TGS, and sends a ticket for TGS.

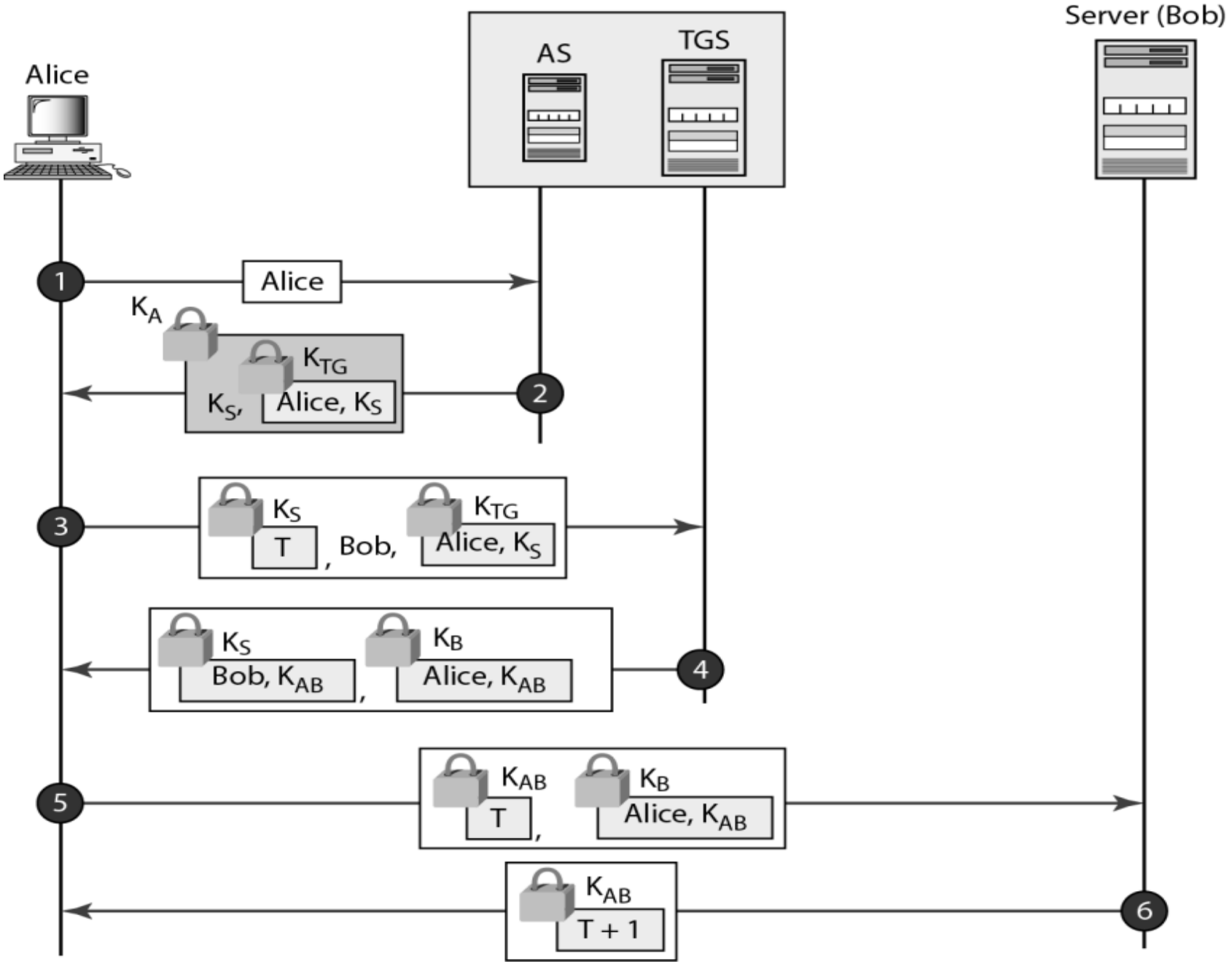
- **Ticket-Granting Server (TGS):** TGS issues a ticket for the real server (Bob). It also provides the session key ($K_{AB)}$ between Alice and Bob. Kerberos has separated the userverification from ticket issuing. In this way, although Alice verifies her ID just once with AS, she can contact TGS multiple times to obtain tickets for different real servers.

- **Real Server:** The real server (Bob) provides services for the user (Alice). Kerberos is designed for a client/server program such as FTP, in which a user uses the client process to access the server process. Kerberos is not used for person-to-person authentication.
- **Operation** :
- Step1. Alice sends her request to AS in plaintext, using her registered identity.

- Step2. AS sends a message encrypted with Alice's symmetric key KA The message contains two items: a session key Ks that is used by Alice to contact TGS and a ticket for TGS that is encrypted with the TGS symmetric key KTG. Alice uses KA to decrypt the message sent. Both Ks and the ticket are extracted.

- Step3. Alice now sends three items to TGS. The first is the ticket received from AS. The second is the name of the real server (Bob), the third is a timestamp which is encrypted by Ks. The timestamp prevents a replay by Eve.
- Step4. Now, TGS sends two tickets, each containing the session key between Alice and Bob KAB. The ticket for Alice is encrypted with Ks; the ticket for Bob is encrypted with Bob's key KB.

- Step5. Alice sends Bob's ticket with the timestamp encrypted by KAB.
- Step6. Bob confirms the receipt by adding 1 to the timestamp. The message is encrypted with KAB and sent to Alice.


Server (Bob)
AS
TGS
Alice
1
Alice
K_A
K_{TG}
K_S,
Alice, K_S
2
3
K_S
T
,
Bob,
K_{TG}
Alice, K_S
4
K_S
Bob, K_{AB}
,
K_B
Alice, K_{AB}
5
K_{AB}
T
,
K_B
Alice, K_{AB}
K_{AB}
T + 1
6

Kerberos: summary

- Authentication method:
  - User's enter password to (AS) only once.
  - Authenticated via central KDC (AS) once per day.
  - No passwords travel over the network.
- Single Sign-on (via TGS):
  - KDC gives you a special "ticket", the TGT, usually good for rest of the day
  - TGT can be used to get other service tickets allowing user to access them (when presented along with authenticators)